Printing

Session 210

Printing

Richard Blanchard

Review: Printing 2002

- A year with major upheaval
  - New printing subsystem
    - Open Source CUPS
    - New printer driver architecture
  - New Print Center
  - Cocoa

Printing 2003

- Evolution of 2002 discontinuities
  - Extend UI to cover more CUPS features
  - Print Center morphs into Desktop printers
  - Extend the power of CUPS printer drivers

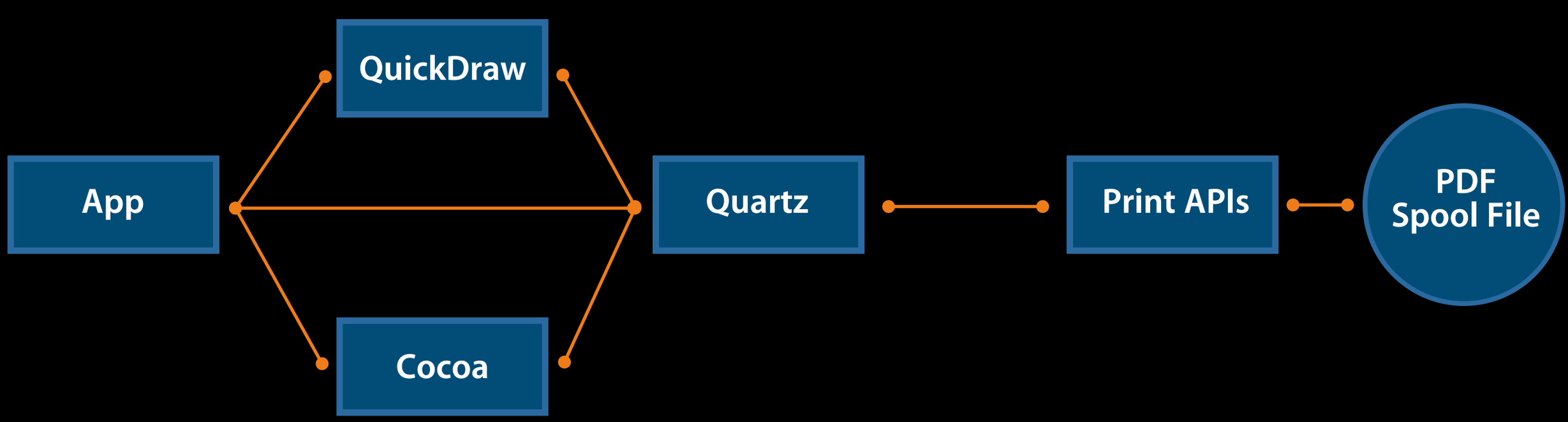
Printing 2003

- Focus on PDF
  - Mac OS X is built around Quartz's PDF imaging model
  - Applications should embrace Quartz to excel
    - Quartz over QuickDraw
    - PDF Printing Path over LaserWriter 8 Legacy Printing Path

PDF Printing Path

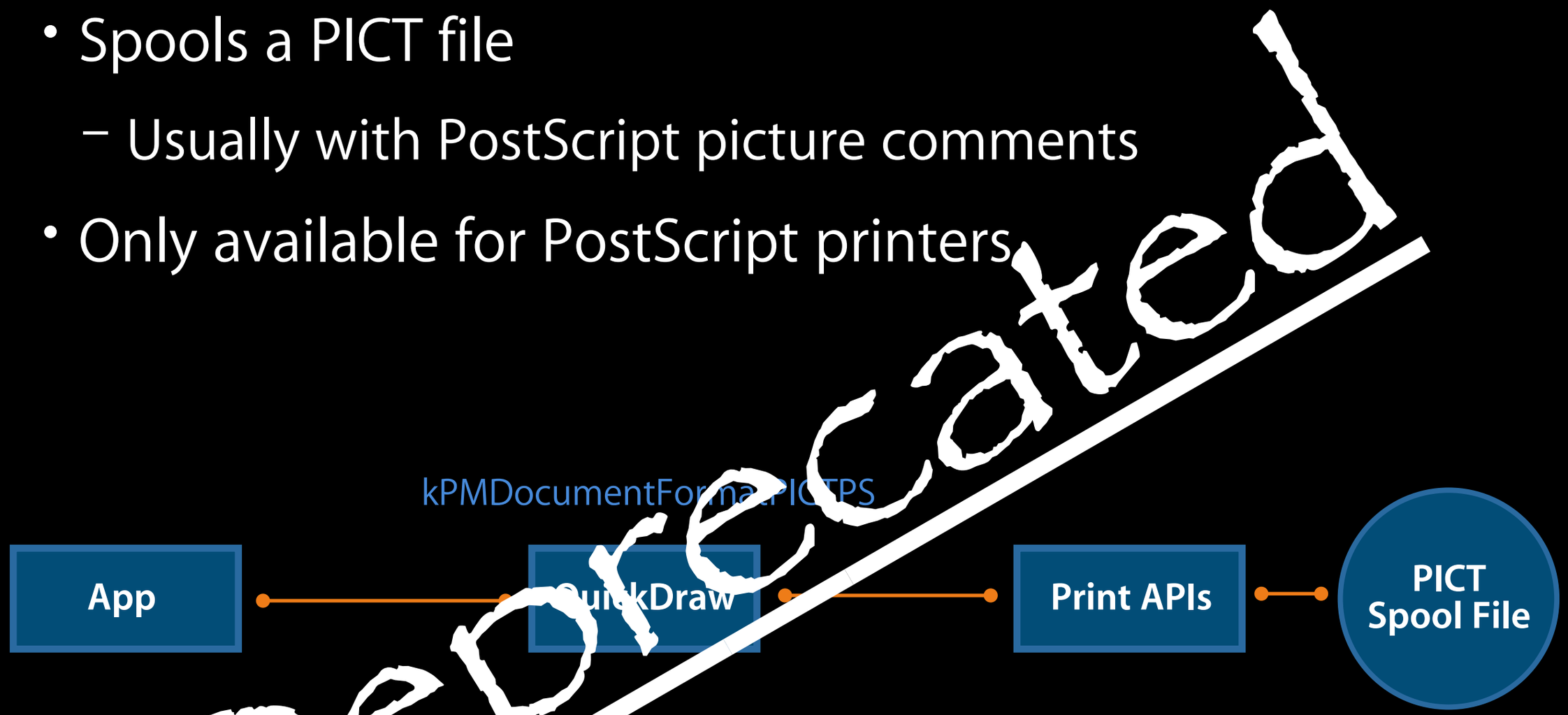
- Spools a PDF file

kPMDocumentFormatPDF



LW8 Legacy Printing Path

- Spools a PICT file
  - Usually with PostScript picture comments
- Only available for PostScript printers



How to Use PDF Printing Path

- The PDF Printing Path is the default path
- Draw with QuickDraw, Cocoa, or Quartz

Going Astray (Danger!)

- Getting on the LW8 Legacy Printing Path takes work:
  - Application must change the format generation
    - /Developer/Examples/Printing/App/BasicPrintLoop
- Beware if you see this code in your app:

```
PMSessionGetDocumentFormatGeneration(printSession,
&supportedFormats);
    …
PMSessionSetDocumentFormatGeneration(printSession,
kPMDocumentFormatPICTPS, arrayOfGraphicsContexts,
NULL);
```

Advantages of the PDF Printing Path

- Why apps should avoid the LW8 Legacy Printing Path
- Application features available on the PDF Printing Path

Advantages of the PDF Path

- Unified Imaging Model
- Preview
- Save As PDF
- PDF Workflow
- Access to Quartz
- Quartz Filters

Advantages: Unified Imaging Model

- PDF spool files can be sent to both PostScript and raster printers
  - Job redirection
  - Application needs only 1 rendering loop
- Digital Master
  - Drawing can be sent to the display, placed on the pasteboard, or sent to the printer

Advantages: Preview

- Laserwriter 8 Legacy Path is not available when the user hits the Preview button
  - Apps must spool PDF
- A good preview has vector PDF
  - Poor substitutes:
    - A rasterized PDF rendition of your document
    - Disabling the Preview function

Advantages: Save As PDF

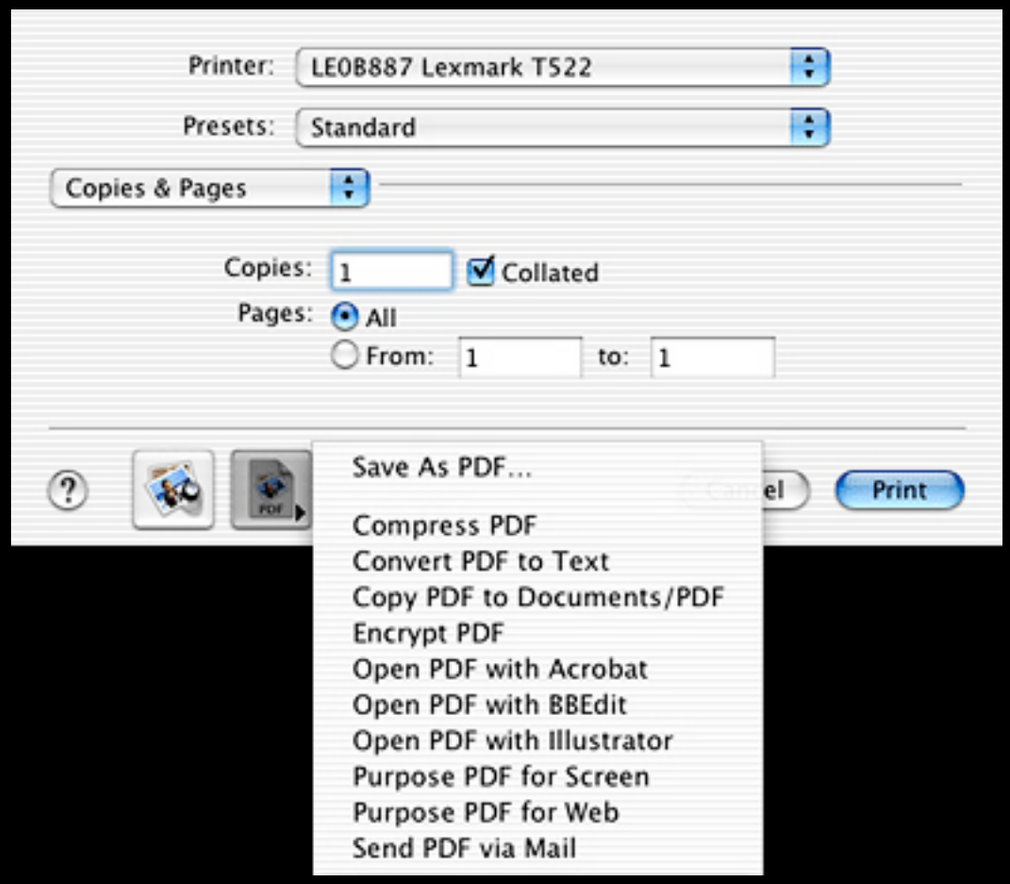
- LW8 Legacy Printing Path is not available
- Digital Master
  - What you draw is what is saved
    - Draw Vectors
    - Draw Efficiently (more later)

Advantages: PDF Workflow

- Released in 10.2.4
- Hooks the user side of the print system
- Process the Digital Master PDF
- Documentation
  - “Providing PDF Workflow Options in the Print Dialog”

PDF Workflow

- Workflow items are placed here:
  - ~/Library/PDF Services, /Library/PDF Services
- Supported Workflow items
  - Folder
  - Folder alias
  - Application or application alias
  - Unix tool
  - AppleScript
  - Quartz Filter (new on Panther)



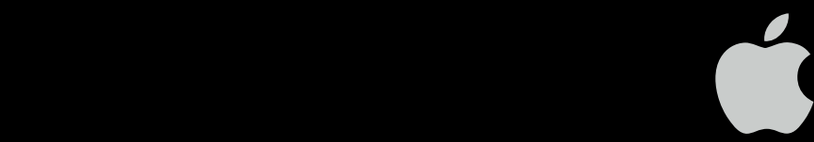
PDF Workflow Tool

- Any shell executable file
  - Compiled tool
  - Interpreted tool
    - Python
    - Perl
    - Ruby
    - Shell Script
- Usage: tool title options pdf-file

PDF Workflow Example

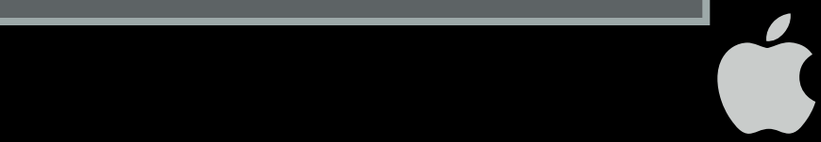
Using Quartz Scripting

```
#!/usr/bin/python
# Create a CMYK TIFF for each page in the PDF

import sys,posixpath
from CoreGraphics import *

if len(sys.argv) != 4 :
  print "Usage: cmykPDFWorkflow title options pdf-file"

else :
      …
```



PDF Workflow
Using Quartz Scripting

```
pdf = CGPDFDocumentCreateWithProvider(
        CGDataProviderCreateWithFilename(sys.argv[3]))

for i in range(1, pdf.getNumberOfPages() + 1) :

    mediaBox = pdf.getMediaBox(i)
    ctx = CGBitmapContextCreateWithColor(
            mediaBox.size.width, mediaBox.size.height,
            CGColorSpaceCreateDeviceCMYK(),
            (0, 0, 0, 0, 1))

    ctx.drawPDFDocument(mediaBox, pdf, i)

    path = "%s/%s.%d.tiff" % (posixpath.expanduser("~/TIFFs"),
                              sys.argv[1], i)

    ctx.writeToFile(path, kCGImageFormatTIFF, "")
```



PDF Workflow Demo

CMYK TIFFs
Translating PDFs

Advantages: Access to Quartz

- QuickDraw and Quartz Drawing can be mixed
  - Easy transition to Quartz for Carbon apps
- Efficient re-use of drawing
  - Images
  - Document fragments

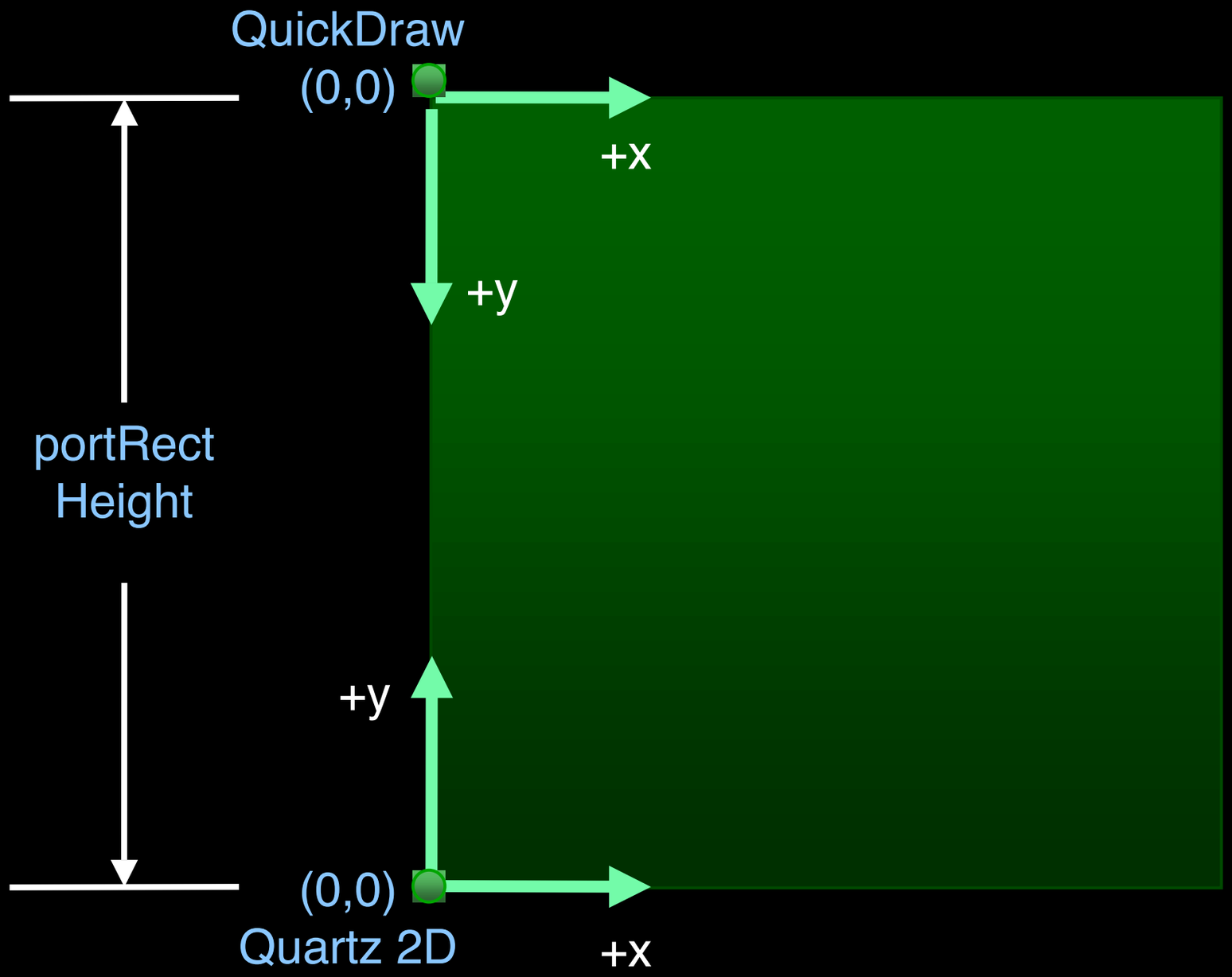
Mixing QuickDraw and Quartz

- Drawing Quartz into a GrafPort

- Available in 10.1 and later
- CarbonQuartzDrawingWPrinting sample code available

Mixing QuickDraw and Quartz

Coordinates

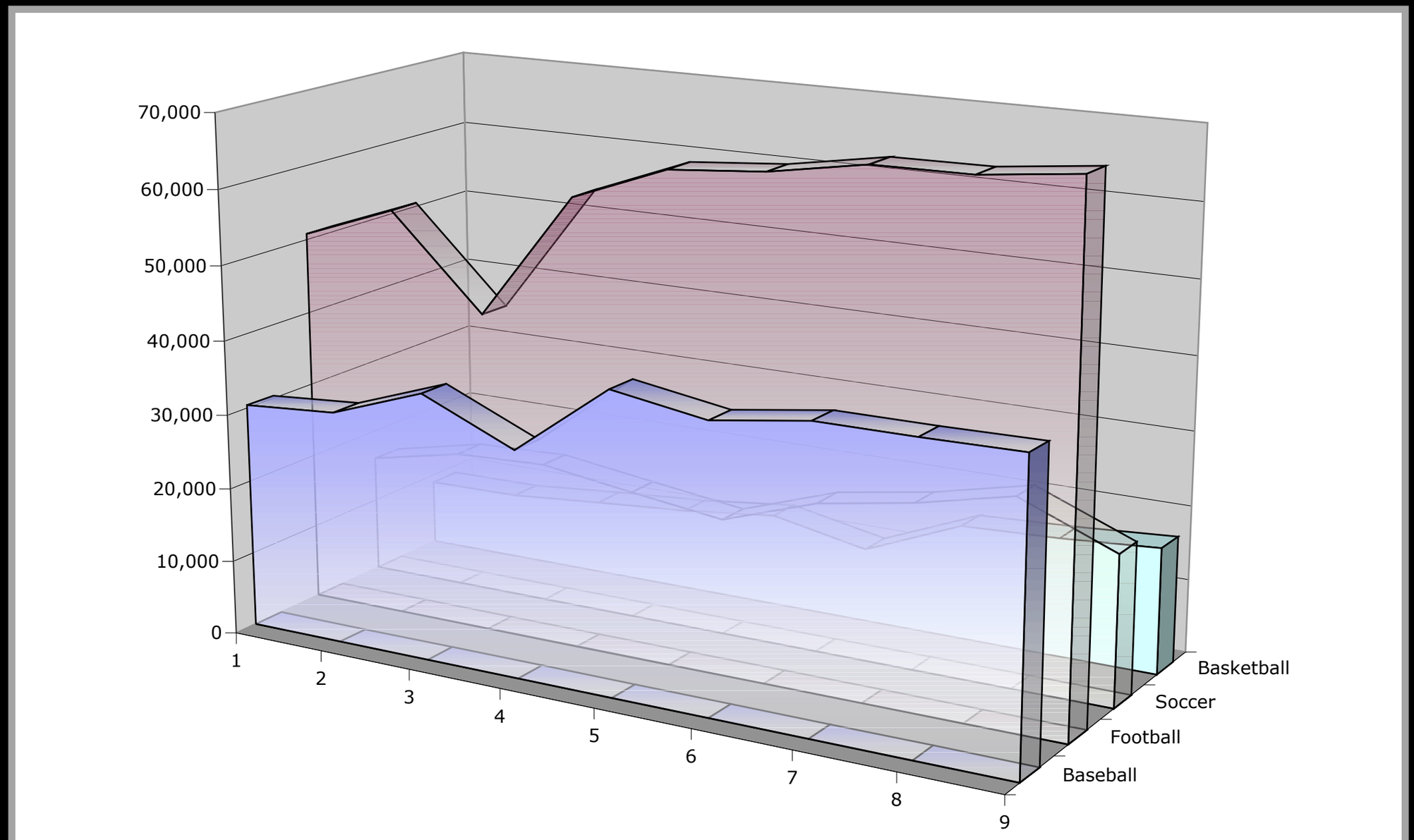


QuickDraw and Quartz

- If QuickDraw and Quartz coordinates must match
  - Perhaps for event handling
- Consider flipping the Quartz y-coordinate
  - Text will draw flipped

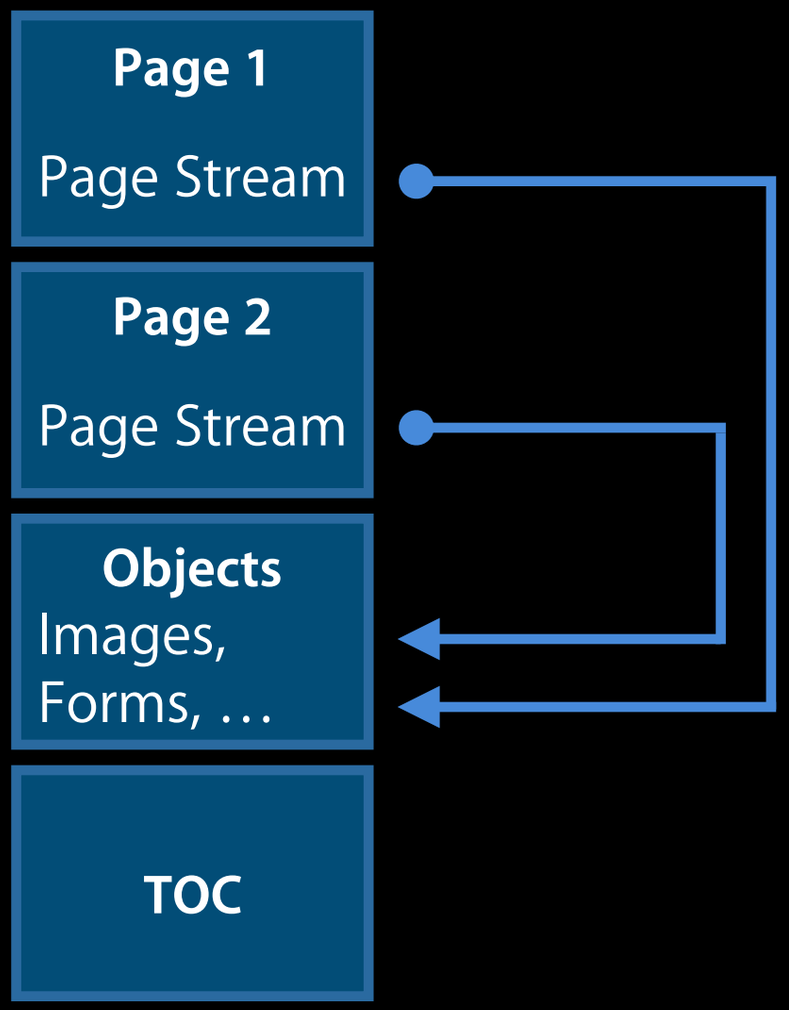
QuickDraw and Quartz Example

Microsoft Excel

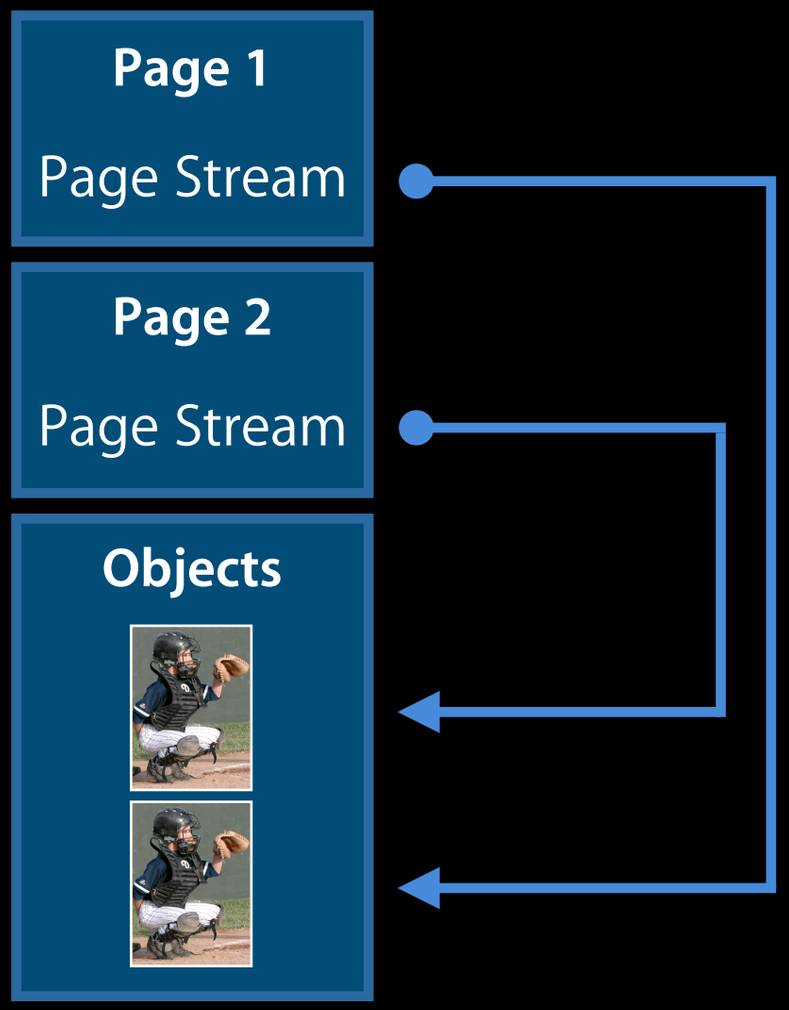


Efficient Drawing

An overview of the PDF format



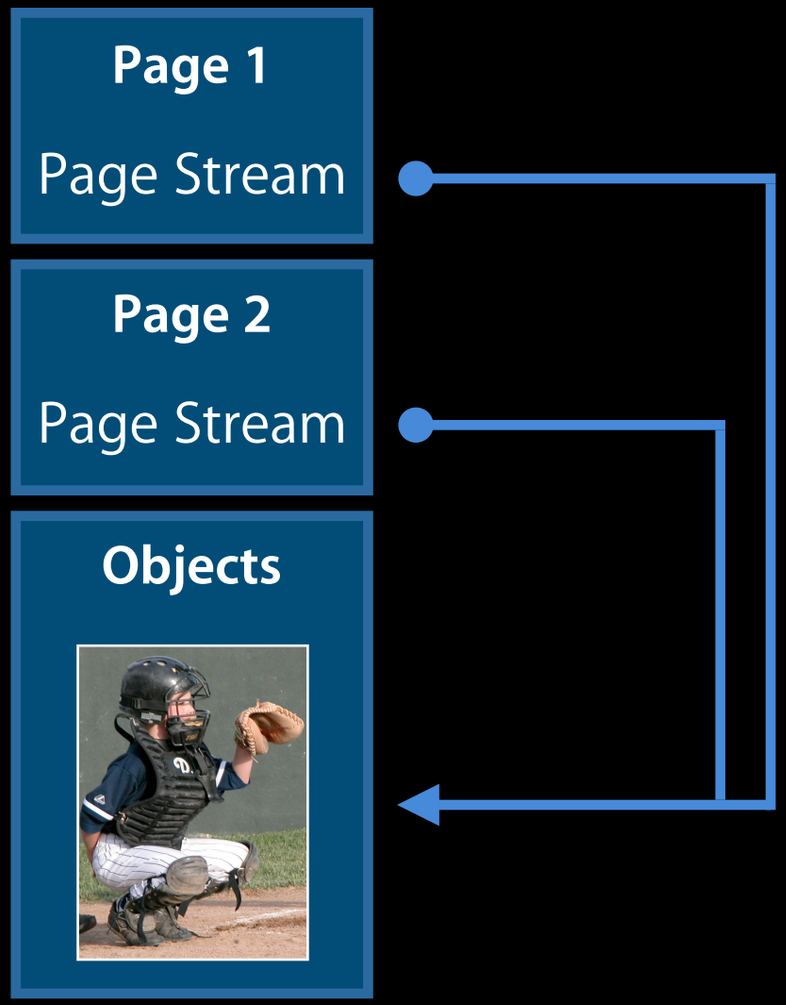
Duplicate Images



Re-Using Quartz Images

- CGImageRefs are easy to create
  - From raw image, JPEG, or PNG data
  - Immutable
  - Use the single image reference multiple times
- This cannot be done in QuickDraw as PixMaps are mutable

Re-Using Images



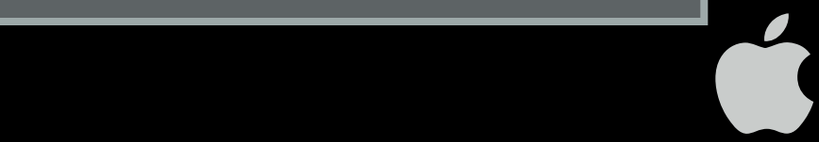
Re-using Images
Quartz Scripting and PDF workflow

```
#!/usr/bin/python
# Watermark each page in a PDF document
import sys, os
from CoreGraphics import *

if len(sys.argv) != 4 :
  print "Usage: watermark title options pdf-file"

else :
  pdfPath = os.path.join(os.path.expanduser("~"),
              "watermarked.pdf")
  imagePath = os.path.join(os.path.expanduser("~"),
                "watermark.jpg")

  defaultBox = CGRectMake(0, 0, 612, 792)
  ctx = CGPDFContextCreateWithFilename(pdfPath,
          defaultBox)
```

Re-using Images

Quartz Scripting and PDF workflow

```
image = CGImageImport(
          CGDataProviderCreateWithFilename(imagePath))
pdf = CGPDFDocumentCreateWithProvider(
        CGDataProviderCreateWithFilename(sys.argv[3]))

for i in range(1, pdf.getNumberOfPages() + 1) :
  mediaBox = pdf.getMediaBox(i)
  imageBox = CGRectMake(
               mediaBox.size.width - image.getWidth(), 0,
               image.getWidth(), image.getHeight())

  ctx.beginPage(mediaBox)
  ctx.drawPDFDocument(mediaBox, pdf, i)
  ctx.drawImage(imageBox, image)
  ctx.endPage()
```

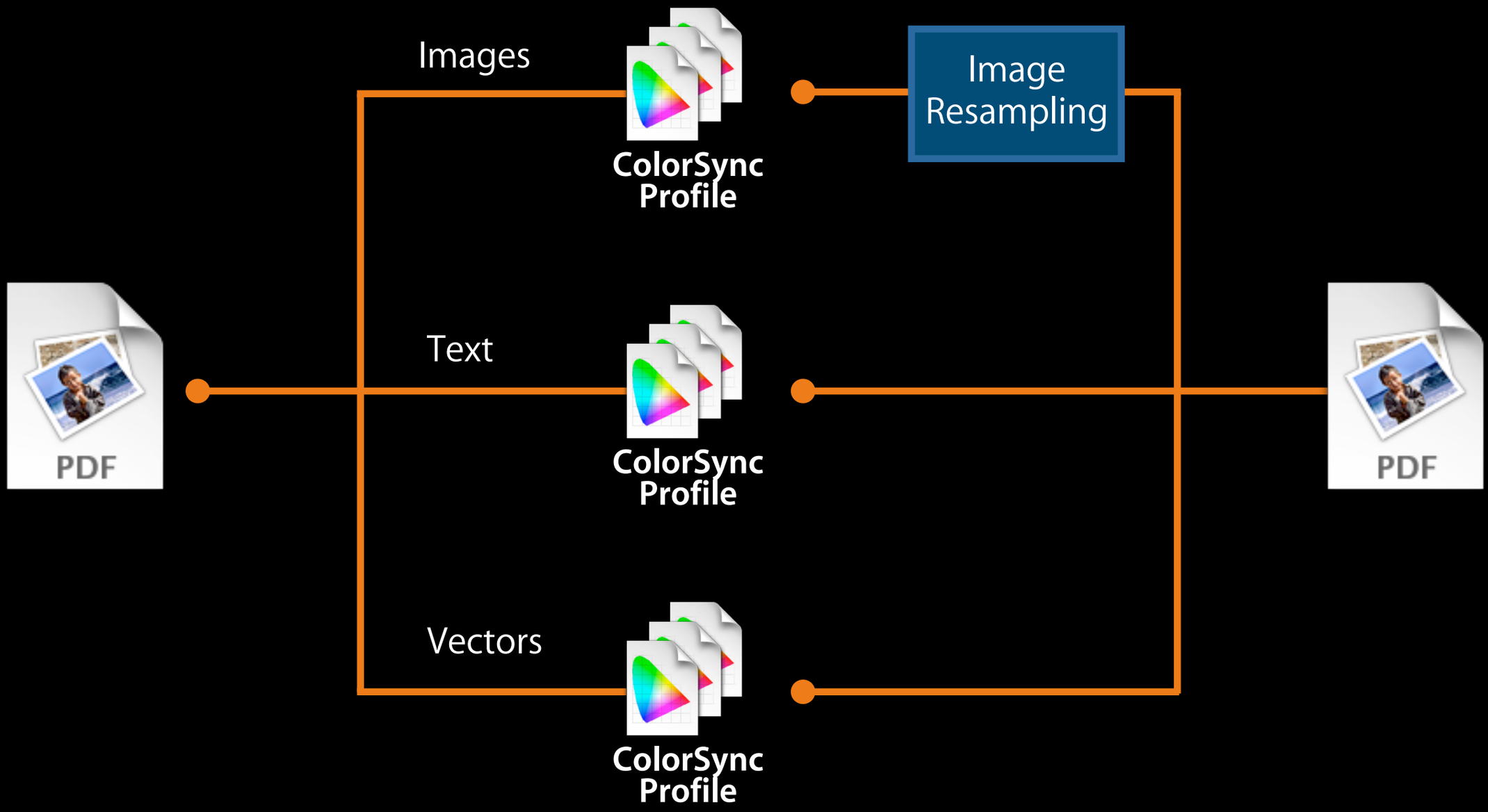


Efficient Drawing

- A similar technique for fragments of vector drawing
  - Logos, headers and footers, etc.
- Capture drawing
  - CGPDFContextCreate or CGPDFContextCreateWithURL
- Create an CGPDFDocumentRef
  - CGPDFCreateWithProvider or CGPDFCreateWithURL
- Draw the fragment where needed
  - CGContextDrawPDFDocument
- Experiment with Quartz Scripting

Advantages: Quartz Filters

What are Quartz filters?



Advantages: Quartz Filters

- Quartz filters can be applied when:
  - Printing
    - Host-based color matching
  - Previewing
  - Saving as PDF
    - PDF Workflow

Quartz Filter Demo

Soft Proofing

Review: Advantages of the PDF Path

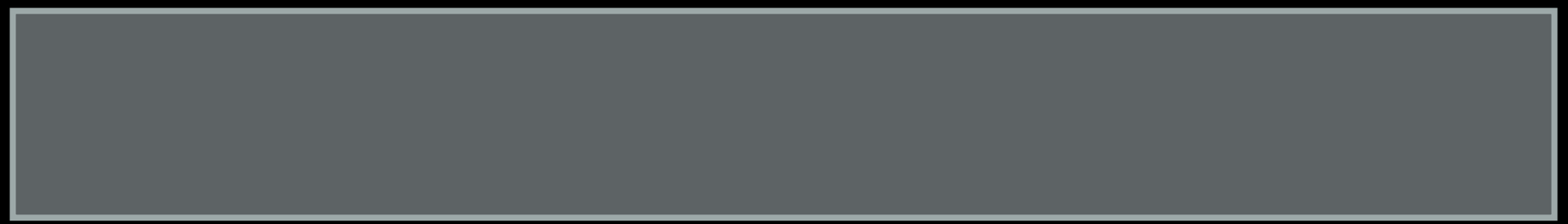
- Unified Imaging Model
- Preview
- Save As PDF
- PDF Workflow
- Access to Quartz
- Quartz filters

Giving Up the LW8 Legacy Path

- You think you can’t because:
  - Your application needs to support EPS
  - Your application generates its own PostScript

Supporting EPS

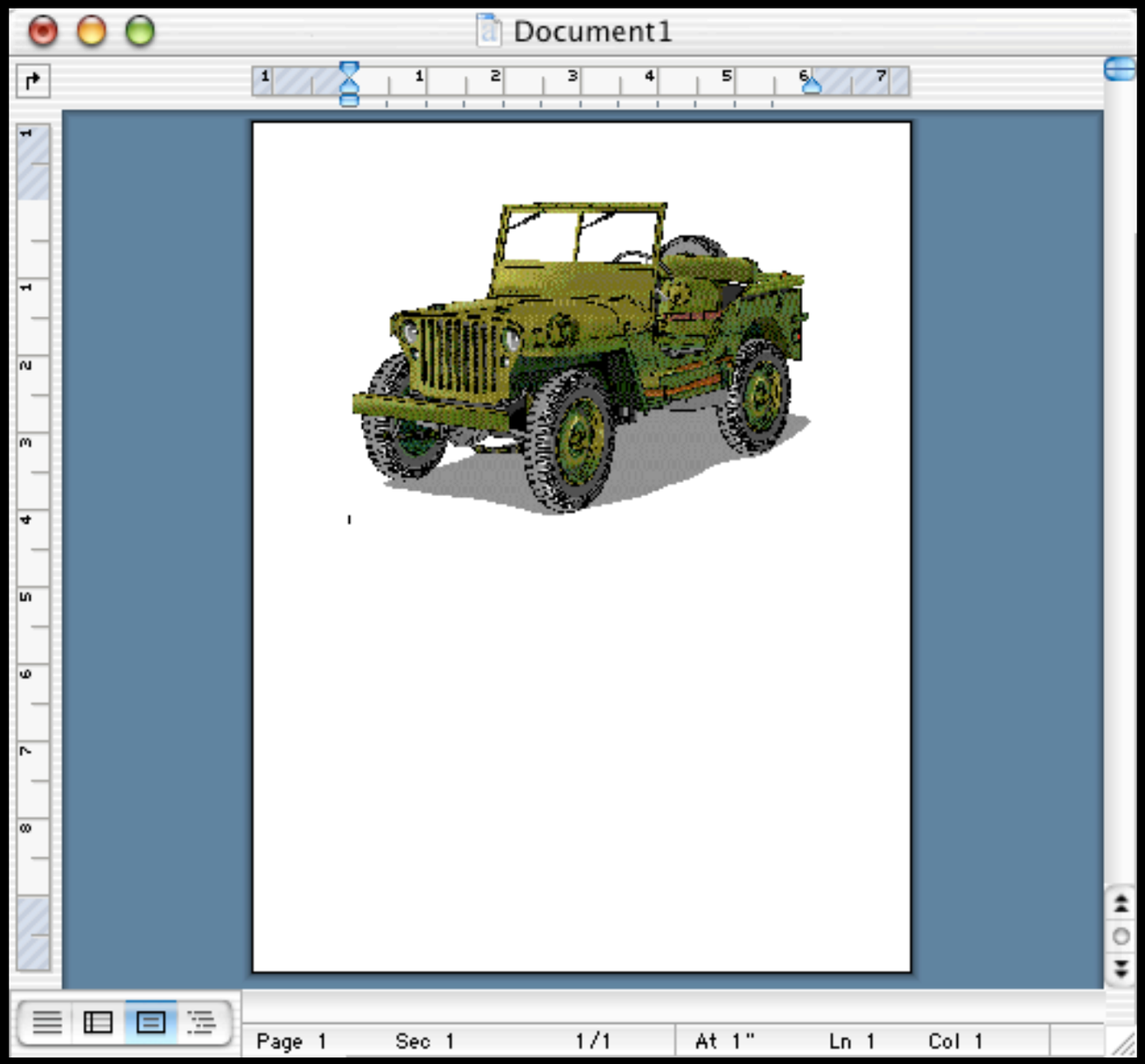
- Use Cocoa
  - Applications get EPS support from NSEPSImageRep
- Carbon apps use:



  - Draw CGImageRef with Quartz
    - The display and raster printers may get the preview image
    - PostScript printers get the EPS PostScript data

Supporting EPS in a Carbon App

Microsoft Word

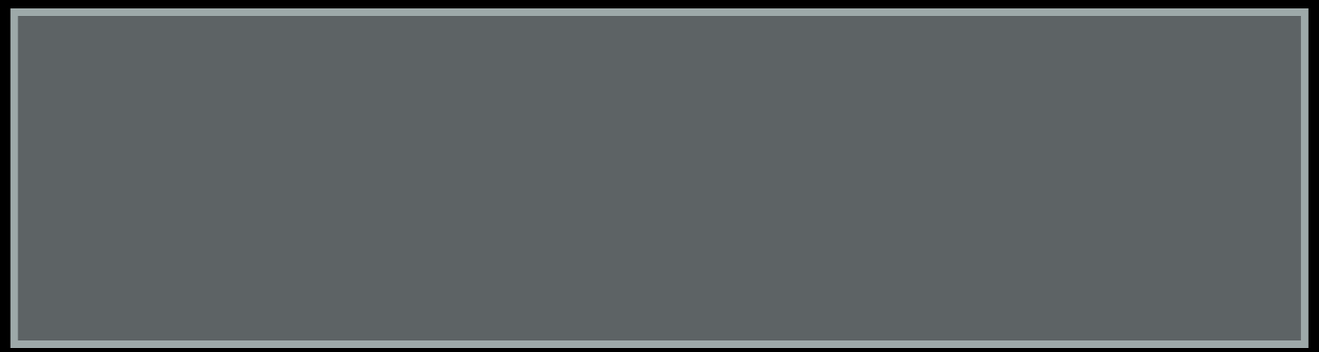


Apps That Generate PostScript

- The choice:
  - LW8 Legacy Printing Path
    - Complex conventions for the interaction of app and driver PostScript streams
    - Space hack, PSBeginNoSave, flushing the QD state
  - Generate the entire PostScript job
- Generating the whole job is better
  - But how to print it?

Job Submission APIs

- Hand a file to the print system
- First check the list of supported mime types



- If the mime type of your file is supported then submit the job

Job Submission APIs

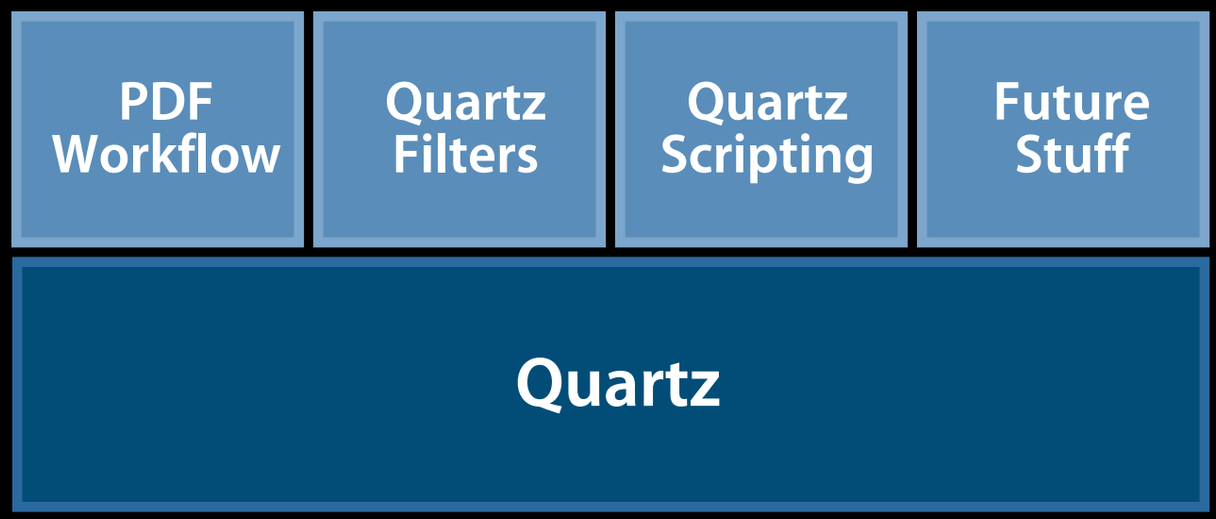
- Mime Type Examples
  - application/postscript
  - application/vnd.cups-postscript
  - application/pdf
  - image/gif, image/jpeg, image/tiff
  - text/plain, text/rtf, text/html
  - application/vnd.cups-raw
  - And more…

Preview

- Uses the Job Submission APIs to print
  - Print previews
  - PDF documents
    - Not in the WWDC seed
- Removes spool time from printing

Review

- Mac OS X has a PDF-based graphics system
  - Use the PDF Printing Path
  - Otherwise investigate the Job Submission APIs
    - Please provide us with feedback
  - Let your app grow with Quartz



Mac OS X Is PDF-Based, But…

- There are important legacy graphics formats
  - PostScript
- It would be great if Mac OS X
  - Rendered EPS files on the display
  - Printed PostScript files on raster printers
  - Converted PS files to PDF files

Panther PostScript Converter

- The PostScript converter can
  - Interpret PostScript Level 3
  - Create PDF files
- The PostScript converter can not
  - Rasterize a PostScript job
- Quartz can rasterize the resultant PDF

PostScript Converter

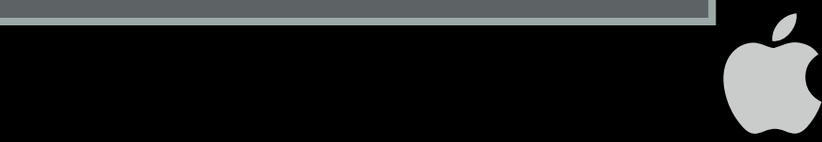
How to write a PS RIP

```
#!/usr/bin/python
# Usage: psrip ps-file tiff-base
#
import sys
from CoreGraphics import *

psProvider = CGDataProviderCreateWithFilename(sys.argv[1])
pdfConsumer = CGDataConsumerCreateWithFilename(sys.argv[2])
converter = CGPSConverterCreateWithoutCallbacks()
success = converter.convert(psProvider, pdfConsumer)

if success :
  pdf = CGPDFDocumentCreateWithProvider(
          CGDataProviderCreateWithFilename(sys.argv[2])
```

PostScript Converter

How to write a PS RIP

```
for i in range(1, pdf.getNumberOfPages() + 1) :
  mediaBox = pdf.getMediaBox(i)
  ctx = CGBitmapContextCreateWithColor(
          mediaBox.size.width, mediaBox.size.height,
          CGColorSpaceCreateDeviceRGB(), (0, 0, 0, 1))

  ctx.drawPDFDocument(mediaBox, pdf, i)
  path = "%s.%d.tiff" % (sys.argv[2], i)

  ctx.writeToFile(path, kCGImageFormatTIFF, "")
```



PostScript Converter Integration

- Cocoa's EPSImageRep converts the EPS to PDF for display
- PostScript jobs can be printed to raster printers
  - Incoming jobs from IPP, LPD, and SMB
    - Great for schools
    - Great for printer vendors
  - Job Submission APIs can send PostScript to raster printers

PostScript Converter Demo

Cocoa EPS Support
Opening a PS File in Preview

New APIs

PMPaper

- Represents the physical characteristics of a PMPageFormat
  - Get the PMPaper from a PMPageFormat:

```
```

  - Get Attributes
    - ID, Name, Height, Width, Margins
  - Get the list of papers for a printer

```
```

PMPaper (Cont.)

- Creating a custom paper size:

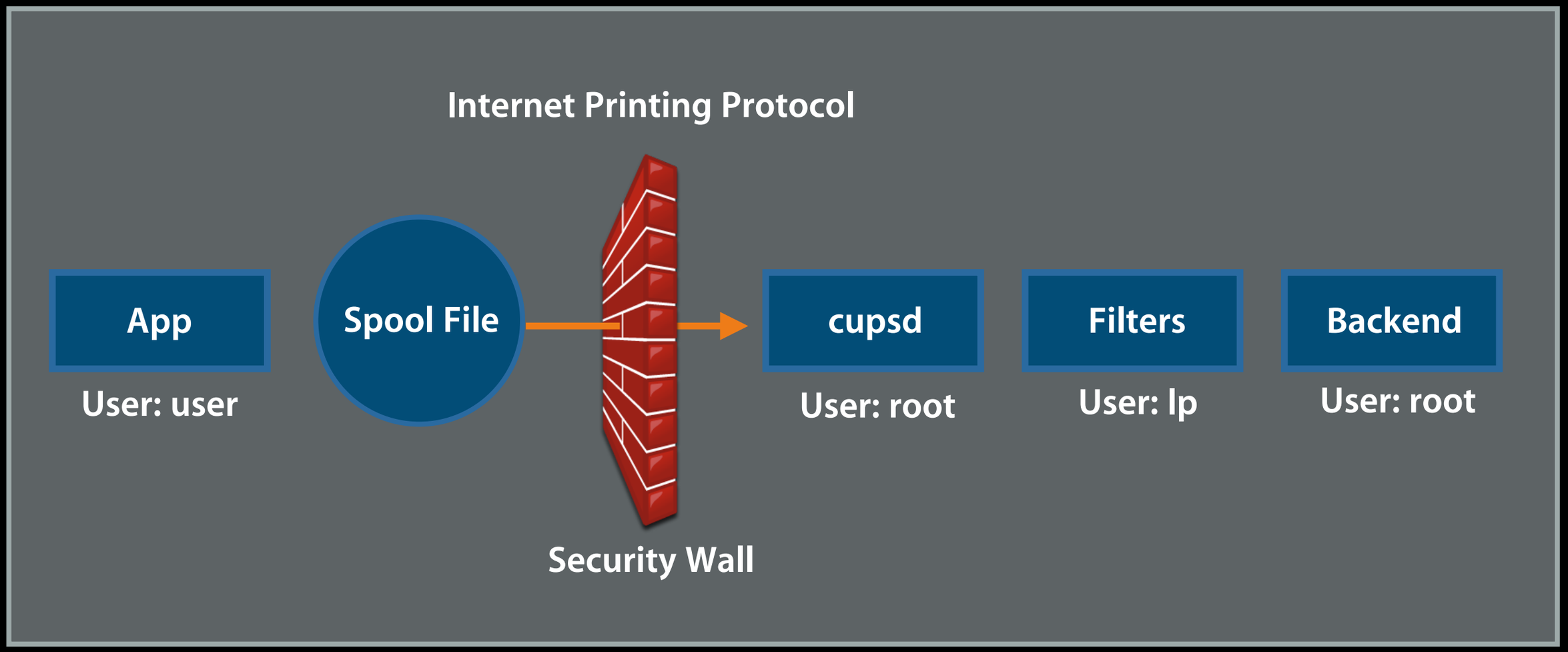
```
```

PPD APIs

- Most applications do not need these functions
  - Enumerate the installed PPDs:
    -
  - Copy a localized version of a PPD:
    -
  - Get uncompressed PPD data:
    -

CUPS Update

What Happens to the Spool File?



CUPS in Mac OS X

- Jaguar was the first Mac OS X OS with CUPS
  - 10.2 was based on CUPS 1.1.15
  - 10.2.x picked up important fixes and security updates from CUPS 1.1.16–1.1.18.
- Panther is based on latest released CUPS
  - 1.1.19
    - Improved PPD support
    - Fax support
    - TBCP Support
    - DoS security fixes

CUPS in Mac OS X

- The Darwin CUPS repository is live
  - Track changes and new features
  - Provide fixes

CUPS in Panther

- Panther will add UI for more CUPS features:
  - SMB printing
  - Direct socket and IPP printing
  - Printer pools
  - TBCP
- Panther adds improved support for CUPS based printer drivers
  - Support for printer dialog extensions
  - ColorSync support
    - Includes direct CMYK rendering

PPD Additions

- For CUPS raster and PS printers
  - ColorSync Profile Reference
    - *cupsICCProfile ColorModel.MediaType.Resolution: <profile-name>
  - Link to Print Dialog Extension
    - *APDialogExtension: <path>
  - Link to Printer Utility
    - *APPrinterUtilityPath: <path>
  - Link to Printer Icon
    - *APPrinterIconPath: <path>

CUPS-Based Printer Drivers

- Going forward, OS support will focus support on CUPS based printer drivers
  - Older Tioga-based printer drivers will continue to be supported
  - Please provide feedback on features needed to better support your CUPS based printer drivers

Open Source Printer Drivers

- Open Source project provides important support:
  - Older printers
  - CUPS-based driver features
- Gimp-Print is in Panther
  - Thanks to Gimp-Print team
  - Thanks to Tyler Blessing’s Mac OS X support

Printing 2004

- Continue PDF Focus
  - Expand features
  - Improve performance
- CUPS 1.2
  - Focus purely on CUPS-based drivers

For More Information

Reference Library

Documentation

About the Mac OS X Printing System

http://developer.apple.com/techpubs/macosx/CoreTechnologies/graphics/Printing/About_MacOSX_Printing/index.html

Supporting Printing in Your Carbon Application

http://developer.apple.com/techpubs/macosx/Carbon/graphics/CarbonPrintingManager/CPM_Concepts/index.html

Cocoa Printing

http://developer.apple.com/techpubs/macosx/Cocoa/TasksAndConcepts/ProgrammingTopics/Printing/index.html

Extending Printing Dialogs

http://developer.apple.com/techpubs/macosx/CoreTechnologies/graphics/Printing/ExtPrintingDialogs/index.html

Reference Library

Documentation (Cont.)

Carbon Printing Manager Reference

http://developer.apple.com/techpubs/macosx/Carbon/graphics/CarbonPrintingManager/CarbonPrintingManager_Ref/cpmref_reference/index.html

Printing Plug-in Interfaces Reference

http://developer.apple.com/techpubs/macosx/CoreTechnologies/graphics/Printing/PrintingPlugin_Ref/ppiref_main/index.html

Using PostScript Printer Description Files

http://developer.apple.com/techpubs/macosx/CoreTechnologies/graphics/Printing/UsingPPDFiles/index.html

Reference Library

Documentation (Cont.)

Creating Presets for iPhoto

http://developer.apple.com/techpubs/macosx/CoreTechnologies/graphics/Printing/PresetDraft/index.html

Providing PDF Workflow Options in the Print Dialog

http://developer.apple.com/techpubs/macosx/CoreTechnologies/graphics/Printing/PDF_Workflow/index.html

CUPS Software Programmers Manual

http://www.cups.org/spm.html

Developer — Mac OS X Printing

http://developer.apple.com/printing/

Reference Library

Source Code

AppUsingSheets

http://developer.apple.com/samplecode/Sample_Code/Printing/AppUsingSheets.htm

CarbonQuartzDrawingWPrinting

http://developer.apple.com/samplecode/Sample_Code/Printing/CarbonQuartzDrawingWPrinti_.htm

PDE Project

http://developer.apple.com/samplecode/Sample_Code/Printing/PDEProject.htm

Roadmap

200 Graphics and Imaging Overview Latest developments in Mac OS X graphics APIs.	**Presidio** **Tue., 10:30am**
201 Mac OS X OpenGL in Depth Introduction to OpenGL on Mac OS X.	**Nob Hill** **Tue., 2:00pm**
202 Technology Magic of Keynote How Keynote uses Mac OS X graphics technologies.	**Nob Hill** **Tue., 3:30pm**
204 Image Capture Update Integrating Image Capture with your product.	**The Marina** **Wed., 9:00am**

Roadmap

205 Vertex Programming with OpenGL Super-fast geometry transformations using the GPU.	**Marina** **Wed., 3:30pm**
206 ColorSync Update Latest developments in Mac OS X color management.	**Mission** **Wed., 5:00pm**
207 Quartz 2D in Depth Harnessing the power of Quartz 2D in your product.	**Presidio** **Thu., 9:00am**
208 Fragment Programming with OpenGL Creating incredible per-pixel effects on Mac OS X.	**Pacific Heights** **Thu., 10:30am**

Roadmap

209 OpenGL Optimizations Obtaining maximum 3D performance on Mac OS X.	**The Marina** **Thu., 2:00pm**
210 Mac OS X Printing Update Latest developments in the Mac OS X printing system.	**Mission** **Thu., 3:30pm**
211 Introduction to Quartz Services Capturing & configuring displays on Mac OS X.	**Pacific Heights** **Fri., 9:00am**
212 Cutting-Edge OpenGL Techniques Expert recipes for stunning 3D effects.	**Russian Hill** **Fri., 2:00pm**

Roadmap

421 Mac OS X Accessibility Latest developments in Mac OS X accessibility.	**Presidio** **Fri., 10:30am**
FF018 Graphics & Imaging Feedback Share your thoughts on graphics & imaging technologies.	**North Beach** **Fri., 5:00pm**

Q&A

Richard Blanchard
Printing

http://developer.apple.com/wwdc2003/urls.html

WWDC 2003

Worldwide Developers Conference